fmLRE: A Low-Resource Relation Extraction Model Based on Feature Mapping Similarity Calculation

Peng Wang[1*†], **Tong Shao**[1†], **Ke Ji**[1], **Guozheng Li**[1], **Wenjun Ke**[1, 2]
[1]Southeast University
[2]Beijing Institute of Computer Technology and Application
pwang@seu.edu.cn, 22S151082@stu.hit.edu.cn, {keji, gzli}@seu.edu.cn, kewenjun2191@163.com

Abstract

Low-resource relation extraction (LRE) aims to extract relations from limited labeled corpora. Existing work takes advantages of self-training or distant supervision to expand the limited labeled data in the data-driven approaches, while the selection bias of pseudo-labels may cause the error accumulation in subsequent relation classification. To address this issue, this paper proposes fmLRE, an iterative feedback method based on feature mapping similarity calculation to improve the accuracy of pseudo-labels. First, it calculates the similarity between pseudo-label and real-label data of the same category in a feature mapping space based on semantic features of labeled dataset after feature projection. Then it fine-tunes the initial model via the iterative process of reinforcement learning. Finally, the similarity is used as a threshold for screening high-precision pseudo-labels and the basis for setting reinforcement learning rewards, which also acts as a penalty term for the relation classifier loss. Experimental results demonstrate that fmLRE achieves the state-of-the-art performance compared with strong baselines on two public datasets.

Introduction

Relation extraction (RE) (Bach and Badaskar 2007) aims to extract semantic relations of given entity pairs and convert unstructured text into the form of triple $(head\_entity, relation, tail\_entity)$. Supervised learning methods (Liang et al. 2022; Li et al. 2022; Zhang et al. 2022; Wu and He 2019), which are based on large-scale datasets and deep neural networks, have shown excellent performance on relation extraction tasks. However, the large-scale and high-quality labeled data is often not available in industrial scenarios, which is called the low-resource relation extraction (LRE) problem (Deng, Zhang, and Chen 2022).

Current LRE work can be roughly divided into two categories: task-oriented and data-oriented. The former combines relation extraction with few-shot learning such as meta-learning (Qu et al. 2020), enables the model to learn information apart from feature representation when performing relation extraction, or to learn novel relations with the help of external knowledge (Levy et al. 2017), pre-trained models (Soares et al. 2019), etc. For instance, Ren et al. (2020) propose the two-stage prototype network with prototype attention alignment and triple loss to improve the generalization of the model, which can dynamically identify new relations with a small number of supporting instances without catastrophic forgetting. However, this method has drawbacks on task generalization and model complexity. To handle these problems, Dong, Pan, and Luo (2021) leverage the idea of metric learning to optimize the LRE problem in the pre-training stage. Peng et al. (2020) also discuss the impact of different semantic feature selections on classification network performance in low-resource situations to reduce the complexity. Therefore, the metric learning is beneficial for better task generalization and less model complexity.

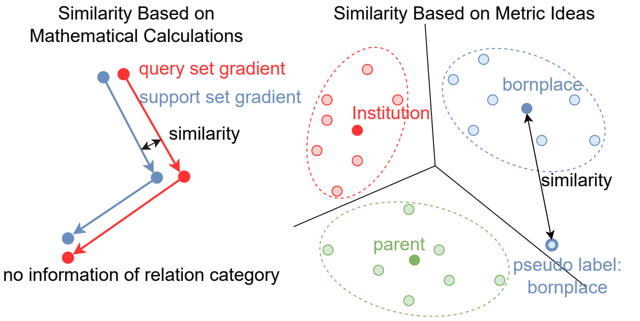

Similarity Based on Mathematical Calculations
Similarity Based on Metric Ideas
query set gradient
support set gradient
similarity
no information of relation category
Institution
bornplace
similarity
parent
pseudo label: bornplace

Figure 1: Comparison with two similarity calculations. The gradient similarity calculation does not consider the relation category information, but calculates the average similarity of the entire dataset. In the right example of metric based similarity, the dotted box is a type of support set feature, and the solid circle is the mean of this type of feature. The predicted pseudo-label `bornplace` is a query sample and the average feature of `bornplace` is the support set. Their distance in the mapping space is the similarity.

For the data-oriented LRE work, a traditional way is distant supervision (Mintz et al. 2009), while it often introduces noise due to the assumption that the same relation must exist between the same entity pair (Hu et al. 2019). To take full advantage of the unlabeled data in the low-resource scenario, Rosenberg, Hebert, and Schneiderman (2005) propose self-training by labeling the unlabeled data to expand the labeled

[*]Corresponding author.
[†]These authors contributed equally.

dataset. However, poor classifier performance trained on few labeled data leads to insufficient precision of the pseudo-label, resulting in selection bias problem (Zhang et al. 2021). Selection bias means that the poor quality of pseudo-label could cause errors to accumulate on the classification model. Therefore, improving the quality of pseudo-label can alleviate the selection bias problem and improve the model performance. To this end, iterative feedback methods are proposed to handle this problem. Hu et al. (2021a) combine iterative feedback with meta-learning to obtain pseudo-labels. However, when the classification network is trained directly by a large number of pseudo-labels, the selection bias inevitably appears. Furthermore, iterative feedback and reinforcement learning are employed to make the gradient of the pseudo-label gradually approximate to the real labels (Hu et al. 2021b), which shows the great potential of reinforcement learning in low-resource relation extraction.

In order to alleviate the selection bias, this paper proposes fmLRE which adopts the idea of metric learning and reinforcement learning iterative process. On the one hand, we argue that metric learning can be used to make self-adaptive evaluation process of pseudo-labels, which is similar to meta-learning to improve selection bias. As shown in Figure 1, compared with the general gradient, fmLRE can add category information and be more adapted to the specific task due to metric feature embedding layers can be pre-trained on task-specific datasets. In fmLRE, the pseudo-label generator is trained with labeled data, and optionally a pre-trained model specialized for relation extraction can be used to further improve model performance. On the other hand, reinforcement learning is used to evaluate the pseudo-labels in the process of iterative pseudo-label generation, and the idea of metric learning like relation network (Sung et al. 2018) is combined to project the labeled data as the support set and the pseudo-labeled data as the query set into the pre-trained mapping space to calculate the feature similarity. Such similarity has two advantages. One is used to filter the high-precision pseudo-labels and then expand the labeled dataset to fine-tune the original model, achieving the positive feedback effect. Another is used as the basis for setting rewards in reinforcement learning and penalty term in relation classifier loss for overall optimization. Moreover, for the first time to our knowledge, we are among the first to utilize prototype network and relation network in metric learning to handle the low-resource relation extraction task.

The main contributions of this paper are summarized as:

- We propose fmLRE, which utilizes a combination of reinforcement learning and metric learning to compute feature similarities under the feature mapping space to filter high-precision pseudo-labels, performing positive feedback iterations and reducing selection bias.
- We improves the pre-trained model in the pseudo-label generator by replacing the commonly used BERT model (Devlin et al. 2019) with the pre-trained model specializing on relation extraction, and propose a filter mechanism to further reduce the proportion of noisy data and improve the stability of the model in the process of eliminating low-precision pseudo-labels.
- Experimental results demonstrate that fmLRE outperforms current state-of-the-art low-resource relation extraction methods on two public benchmarks of SemEval 2010 Task 8 and TACRED.

Related Work

Relation Extraction: Current relation extraction methods can be divided into unsupervised learning, semi-supervised learning, and supervised learning methods. The unsupervised methods are mainly based on clustering, such as the Rel-LDA1 model (Yao et al. 2011), which considers the relations as topics and constructs a distribution of relation categories. The semi-supervised methods are based on the bootstrap idea (Gupta, Roth, and Schütze 2018), and the classic model is the DIPRE (Brin 1998). Data augmentation methods are often used to improve the semi-supervised relation extraction, such as the label propagation model (Chen et al. 2006). Early supervised relation extraction methods utilize feature-based methods (Xu, Mou, and Li 2015) or kernel-based methods (Pawar, Palshikar, and Bhattacharyya 2017). After the emergence of deep neural networks, CNNs (Lin et al. 2016), RNNs, LSTMs (Hochreiter and Schmidhuber 1997) and hybrid neural network structures are combined to improve the supervised relation extraction models. Kim (2014) proposes a CNNs-based relation extraction model with multilayer filter as well as a maximum pooling. Zhang and Wang (2015) combine the mechanism of bidirectional RNNs to improve the relationship extraction performance. Xu, Mou, and Li (2015) construct a relation extraction model using LSTMs.

Low-resource Relation Extraction: Current LRE work can be roughly divided into distant supervision based methods, semi-supervised learning based methods, meta-learning based methods, and external knowledge based methods. The idea of distant supervision was first introduced by Mintz et al. (2009), and later more work focused on alleviating noise affections, such as RESIDE (Vashishth et al. 2018). The in-depth self-training idea on low-resource problems based on semi-supervised learning mainly introduces the pre-training process into traditional methods to further improve the performance, such as GradLRE (Hu et al. 2021b) which uses the gradient similarity to evaluate pseudo-labeling in real time and improve the performance. Hu et al. (2021a) use the idea of combining LRE with meta-learning to mitigate the noise problem caused by pseudo-labeling. In meta-learning methods, the idea of metric learning is usually used, such as Peng et al. (2020), while the limitation of this method is that at least one data of each relation category needs to be used as a support set, so the metric learning methods cannot work well for zero-shot learning or low-resource problems with extreme lack of data. External knowledge can also be utilized to LRE, such as Cetoli (2020) to transform the LRE task into the question-and-answer task.

Methodology

Problem Statements

This paper explores relation extraction in the low-resource scenarios and aims to extract the relational triples in the form

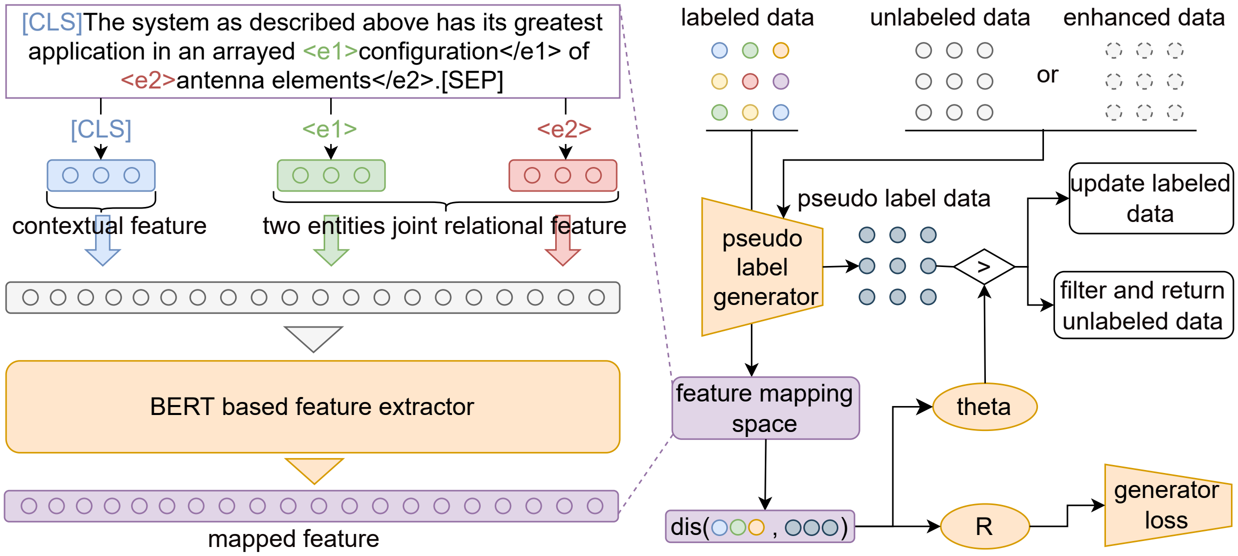

[CLS]The system as described above has its greatest application in an arrayed <e1>configuration</e1> of <e2>antenna elements</e2>.[SEP]
[CLS]
<e1>
<e2>
contextual feature
two entities joint relational feature
BERT based feature extractor
mapped feature
labeled data
unlabeled data
or
enhanced data
pseudo label data
pseudo label generator
>
update labeled data
filter and return unlabeled data
feature mapping space
theta
dis(,)
R
generator loss

Figure 2: The framework of fmLRE. We iteratively generate pseudo-labels for unlabeled data. In order to evaluate the quality of pseudo-labels, the distance between them and support relation prototypes is calculated as the similarity in the pre-trained feature mapping space. The similarity is used to filter high-precision pseudo-labels and update the labeled dataset, and it is also used as the basis for setting rewards in reinforcement learning and penalty term in relation classifier loss.

of $< e1, r, e2 >$ with few labeled data. The whole dataset is denoted as X and a sample in the dataset is represented as x. Following previous work (Soares et al. 2019), the entity pair needs to be given in advance and labeled with marks. In addition, special tokens also needs to be added to the text sentences, where $[\text{CLS}]$ is used to obtain the feature representation afterwards and $[\text{SEP}]$ is used for the sentence separation. The final form of the sample sentence can be shown with an example in Figure 2. Similarly, the whole set of labels in X is denoted as Y, the label of sample x is denoted as y in the dataset, and the generated pseudo-label for the unlabeled sample $\tilde{x}$ is denoted as $\tilde{y}$.

Framework Overview

The framework of fmLRE is shown in Figure 2. Based on the idea of self-training, pseudo-label is applied to the existing unlabeled data so as to expand the labeled dataset and transform the LRE problem into a normal supervised relation extraction task. Due to the selection bias, fmLRE adopts a combination of reinforcement learning and metric learning to provide positive iterative feedback.

Pseudo-Label Generator

The pseudo-label generator acquires feature representations of sentences based on pre-trained models, where specifically the hidden layer representations of head and tail entities are concatenated as the features for sentence classification. Let $\mathbf{h}_{[k]}$ be the hidden layer when it is a k-th token, then the feature of each sentence is represented as $\mathbf{h} = [\mathbf{h}_{[e1]}, \mathbf{h}_{[e2]}], \mathbf{h} \in \mathbf{R}^{2l_{\mathcal{R}}}$, where $l_{\mathcal{R}}$ denotes the length of the feature representation, $\mathcal{R}$ denotes all possible categories of relations. We design a simple classification network, including a hidden layer and a fully connected layer for classification. In this paper, the classification network is considered as a functional form f. Then the predicted pseudo-label expression is as follows:

$$\tilde{y} = \text{argmax}(f(\mathbf{h})) \tag{1}$$

Here the one with the highest probability is selected as the pseudo-label for prediction. And in this classification process, the traditional cross-entropy loss is used for the loss function setting. N indicates the number of all samples in the dataset, and then the predicted pseudo label expression is as follows:

$$\text{loss}(y_i, \tilde{y}_i) = \frac{1}{N} \sum_{i=1}^{N} \text{CrossEntropy}(y_i, \tilde{y}_i) \tag{2}$$

Feature Mapping Space Based on Metric Learning

The training idea of fmLRE is to use the feature representation of labeled data to pre-train a classification network similar to a pseudo-label generator. The fully connected layer of the network is mapping the sample sentence features into a specific space and making simple classification prediction. In other word, the hidden layer state of the fully connected layer is considered as the feature after the projection of the original sample features through the mapping space. Finally, projected features are used in the similarity calculation.

Firstly, the feature representation needs to be selected. In order to enhance the adaptability to low-resource problems, we use both relational and contextual representations g_{rel} and g_{con} as feature representations for training, where the feature representation in the mapping space is denoted as g. Then the input sample feature representation is:

$$g_{\text{rel}}(x, \mathbf{h}_{[e1]}, \mathbf{h}_{[e2]}), g_{\text{con}}(x, \mathbf{h}_{[\text{CLS}]}) \tag{3}$$

In the feature representation composition, relation information is the hidden state of the head and tail entities. For the context representation of the whole sentence, since it is mainly used for classification here, we use the hidden layer of the [CLS] as the context representation of the relation classification. The final feature representation is:

$$\mathrm{feature}(x) = [g_{\mathrm{rel}}(x, \mathbf{h}_{[e1]}, \mathbf{h}_{[e2]}); g_{\mathrm{con}}(x, \mathbf{h}_{[\mathrm{CLS}]})] \quad (4)$$

The projection process of the mapping space can be regarded as a function $F(x)$. The features in the mapping space can be obtained after transforming the feature representations of the corresponding sample sentences, which is then expressed as:

$$G(x) = F(\mathrm{feature}(x)) \quad (5)$$

After obtaining the features in the mapping space, the similarity of the metric learning can be computed. The accuracy of the pseudo-label can be evaluated and the loss function of the pseudo-label generator can be improved.

Reinforcement Learning Iterative Feedback

For unlabeled data divided by the batch, after obtaining the feature representations in the pre-trained mapping space, the similarity between the support set and the query set can be calculated. On the one hand, the similarity is used to evaluate the confidence of the pseudo-label to expand the original dataset. On the other hand, the similarity can be used to set different rewards to improve the pseudo-label generator.

Feedback on Pseudo Labels Let the set of real labeled data with the same pseudo-label be denoted as $X_{\tilde{y}}$. With the pre-trained mapping space, we can obtain the corresponding feature representations as:

$$\bar{G}(X_{\tilde{y}}) = \frac{1}{N_{\tilde{y}}} \sum_{j=1}^{N_{\tilde{y}}} G(x_j) \quad (6)$$

For the real labeled dataset, we first obtain its average features, and then calculate the similarity s between the support set and the query sample:

$$s = \mathrm{similarity}(\bar{G}(X_{\tilde{y}}), G(\tilde{x}), \tilde{y}) \quad (7)$$

Here, $s \in [0, 1]$, and s is also important for judging the confidence of pseudo-label. We set a hyper-parameter θ as the similarity threshold. If the similarity is greater than θ, then it is considered that the query sample and the support set are the same type. The updated true label dataset is used to train the pseudo-label generator with a second fine-tuning. The expanded data is used to facilitate the model fitting to it for improving the performance of the pseudo-label generator. After obtaining the high-precision pseudo-label, it updates the true label dataset and synchronizes the average features of the support set used for similarity calculation with:

$$\bar{G}(X_{\tilde{y}}) = \frac{\bar{G}(X_{\tilde{y}}) * N_{\tilde{y}} + G(\tilde{x})}{N_{\tilde{y}} + 1}, N_{\tilde{y}} = N_{\tilde{y}} + 1 \quad (8)$$

Feedback on Pseudo-Label Generator To improve the loss function in the pseudo-label generator, we use the similarity between the support set and the query samples to set the reward in reinforcement learning. If the similarity between a pseudo-label and the support set is high, it means that the consistency between real label and pseudo-label is high. Therefore, reward R can be set for the threshold θ set above, which indicates a positive feedback. A reward less than 1 can be set when $R > \theta$ and a reward greater than 1 when it is less than θ. The $range$ is also set to control the specific range of the reward, and it is added to the original loss function of the pseudo-label generator in the form of weights:

$$L = \sum_{t=1}^{T} \mathrm{loss}(y, \tilde{y}) * R^{(t)} \quad (9)$$

where t denotes the time step of reinforcement learning and T is the total number of sequences.

Noise Filtering Mechanism For low-precision pseudo-labels eliminated in iterative feedback, there are two error cases: (1) incorrect predictions in known relation types; (2) the presence of noise or unknown relation types. The first case is the problem that the model itself should solve. For the second case, fmLRE takes a special process to deal with it, as shown in Algorithm 1. We limit the number of eliminations of pseudo-labels. If the pseudo-labels do not reach the similarity threshold, we first delete the pseudo-labels and return to the unlabeled data set, and then record the number of eliminations. If the number of eliminations reaches the threshold, the unlabeled data is regarded as noisy data or unknown relation type data, and is deleted from the entire dataset. That can improve the impact of noise on the model as much as possible. Even the strategy of adaptive learning is a better choice, it is more difficult for integer. Therefore, fmLRE uses a simple and effective strategy.

Algorithm 1: Filtering mechanism

```
Input: $l\_dataset$, $ul\_dataset$
Output: Updated $l\_dataset$, $ul\_dataset$
$filer\_flag = [0] * number\_of\_feedback$
Sample $ul\_data\_i$ from $ul\_dataset$
while $iterative\ feedback\ process$ do
    $p\_l \leftarrow pseudo\_label\_generator(ul\_data\_i)$
    $s\_f \leftarrow mean\_feature(l\_dataset[p\_l])$
    $q\_f \leftarrow feature(ul\_data\_i)$
    $s \leftarrow similarity(s\_f,\ q\_f)$
    if $s > \theta$ and $filter\_flag[i] < n$ then
        $ul\_data \rightarrow l\_dataset$
        break
    else if $filer\_flag[i] > n$ then
        Delete $ul\_data\_i$ from $ul\_dataset$
    else
        $ul\_data\_i \rightarrow ul\_dataset$
    end if
end while
Update $pseudo\_label\_generator$ based on $l\_dataset$
return $ul\_dataset$, $l\_dataset$
```

Dataset	SemEval	TACRED	FewRel1.0
Train	8,000	68,124	44,800
Test	2,717	15,509	44,800
# of relation	19	42	100

Table 1: Statistics of datasets.

Deepening Module

There are two components in the deepening module. The first one is data augmentation. When there is a lack of unlabeled data in the low-resource scenario, we use the setting in GradLRE (Hu et al. 2021b) to perform data augmentation and unify the model input, which masks the sample sentences with a certain span. The span can be set as a hyperparameter, which is a proportion of the sentence length and usually sampled from a given data distribution. However, this data augmentation often has difficulties in data diversity, so its performance may lower than that with unlabeled data. The second component is the replacement of the pretrained model. fmLRE uses the MTB (Soares et al. 2019) model for the relation extraction task to replace the BERT, which can further improve the performance of the model. In order to ensure the fairness of the performance comparison, the MTB is not used for comparison with baselines.

Experiments

Datasets and Settings

To evaluate the proposed fmLRE model, we select two public datasets to conduct experiments: SemEval 2010 Task 8 dataset (Hendrickx et al. 2010) and the TACRED dataset (Zhang et al. 2017). We also evaluate the generalization of fmLRE on the FewRel 1.0 dataset (Han et al. 2018). The detailed statistics of the datasets are shown in Table 1. Considering the low-resource scenario, fmLRE only splits a small ratio of the data as labeled dataset, and the rest is used as the unlabeled dataset, which follows common experimental setups: 5%, 10%, and 30% of the data are split into labeled data on SemEval, and the rest is used as unlabeled data. On the TACRED dataset, 3%, 10%, and 15% of the data are split as labeled data.

In the implementation of fmLRE, the max length of the sample sentences is 128, the batch size for training and division of unlabeled data is 32, and the initial learning rate of the model is 5e-5. In addition, there are some hyperparameters: the threshold θ of similarity is set to 0.7, the constraint $range$ of reward is set to 0.2, and the number of filtering for noisy data n is set to 2. The code of fmLRE and the datasets can be accessed via https://github.com/seukgcode/fmLRE.

Baselines

We choose strong LRE methods as baselines.
Self-train (Rosenberg, Hebert, and Schneiderman 2005): It proposes the idea of self-training to generate pseudo-labels and expand the labeled dataset.
MRefG (Li and Qian 2020): It constructs semantic connections between labeled and unlabeled data via citation graphs.
GradLRE (Hu et al. 2021b): It improves the self-training via calculating the gradient similarity between labeled and unlabeled data to reduce the selection error of the model.
MetaSRE (Hu et al. 2021a): It uses a classifier as an additional meta-learning target and then generates pseudo-labels, which is among the best reported models.
UREVA (Yuan and Eldardiry 2021): It utilizes an unsupervised learning with a variable auto-encoder, breaking the restriction that the model needs to follow a prior distribution.
BERTw.gold labels: It is the upper performance of a model that can be obtained from the self-training by bringing the truth labels of unlabeled data into the training process. We call it the gold model in the evaluation.

Moreover, in order to evaluate the generalization, we also compare fmLRE with strong few-shot learning models.
IncreProtoNet (Ren et al. 2020): It uses a two-stage prototype network to dynamically identify novel relations by a few support instances without catastrophic forgetting.
HCRP (Han, Cheng, and Lu 2021): It uses a combination of contrastive learning and hybrid prototype networks to handle fine-grained relation extraction.
MapRE (Dong, Pan, and Luo 2021): It is a pre-trained method based on metric learning and fine-tuning on the few-shot task to deal with relation extraction.

Main Results

Main experimental results are shown in Table 2, indicating that fmLRE achieves the best performance with all ratios of labeled data. On the SemEval, fmLRE achieves the best performance and is close to the gold model. Note that the performance gap between different models is illustrated clearly in Figure 3. In 5% ratio, fmLRE is 2.5% higher than the optimal baseline GradLRE, and only 2.5% away from gold model. In 10% ratio, due to the increase in the scale of labels, the model performance is also enhanced to a certain extent, which nearly 2%. And the performance gap between the gold model and fmLRE is about 2%. Furthermore, in 30% ratio, the advantages of the low-resource models have not been reflected, and fmLRE has only a 0.5% advantage. On the TACRED, fmLRE has the best performance, but the performance gap between it and the gold model is bigger. With the increase on the ratio of labeled data, the performance gap between fmLRE and gold model is constantly narrowing meanwhile maintaining the optimal performance, from more than 10% in 3% ratio to less than 5%. We believe that the scale of labeled data and the proportion of unknown relations affect the performance of the model and the performance gap based on the characteristics of TACRED.

Moreover, there is a high correlation between the evaluation of pseudo-labels and the classification results. Actually the selection bias can be seen from the accuracy of the pseudo-label, that is, the classification performance of the model. The above results demonstrate that fmLRE can improve the selection bias problem caused by the insufficient accuracy of the pseudo-label. Note that the effect with data augmentation is generally lower than that of the orig-

Dataset	SemEval(%)			TACRED(%)		
Ratio	5%	10%	30%	3%	10%	15%
Self-train	71.34	74.25	63.87	42.11	54.17	56.52
GradLRE	79.65	81.69	85.52	47.37	58.20	59.93
MEefG	75.48	77.96	83.24	43.81	55.42	58.21
MetaSRE	78.33	80.09	84.81	46.16	56.95	58.94
UREVA	77.46	78.44	82.71	-	-	-
BERTw.gold labels	84.64	85.40	87.08	62.93	63.66	64.69
fmLRE	**82.11**	**83.83**	**85.97**	**50.83**	**58.94**	**60.62**
fmLRE(data augmentation)	80.04	80.77	82.32	46.21	52.88	56.83

Table 2: Main experimental results. The results are F1 scores, and the bold ones are the best results.

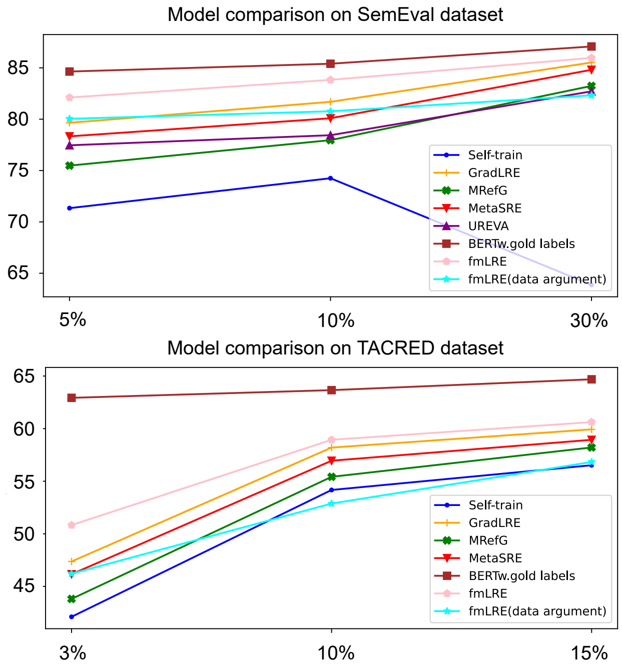


Figure 3: Comparison of performances on SemEval and TACRED datasets.

inal model. In low-resource scenarios, if we have few labeled data, there is definitely deviation from the real distribution due to the limitations of data size and diversity. Consequently, the unlabeled data obtained by data augmentation based on these labeled data tends to have poor diversity as it has high similarity among each other, which is also a general drawback of data augmentation. Therefore, the performance of fmLRE with data augmentation does not perform as well as the original fmLRE.

Generalization Experiments

In view of the relevance between few-shot learning and low-resource problem and the comprehensiveness of the experimental study, we also evaluate the generalization of fmLRE. In the comparison, we uses the N-way K-shot method of few-shot learning as the test strategy, that is, there are N categories each time, and each category has K samples. The experimental results are shown in Table 3. Note that fmLRE only perform well in the prediction performance of the novel relations, but there is a performance gap compared with the current few-shot learning models. Few-shot learning is applicable to rare-data scenarios, while the low-resource problem is more general, including few-shot learning. However, there are still differences between two tasks. Few-shot learning adopts the episode training strategy to learn the commonality of similar tasks, but low-resource problem is more inclined to adopt semi-supervised methods which based on data. Due to the fmLRE is aimed at the low-resource problem with the limited relations and has no special processing mechanism for novel relations, it is reasonable to have a gap comparing with the few-shot learning methods.

Strategy	5w1s	5w5s	10w1s	10w5s
InrceProtoNet	82.10	84.64	-	-
HCRP	90.90	93.22	84.11	87.79
MapRE	**95.73**	**97.84**	**93.18**	**95.64**
fmLRE	84.43	88.88	87.51	88.11

Table 3: Generalization comparison experiment on FewRel 1.0 dataset.

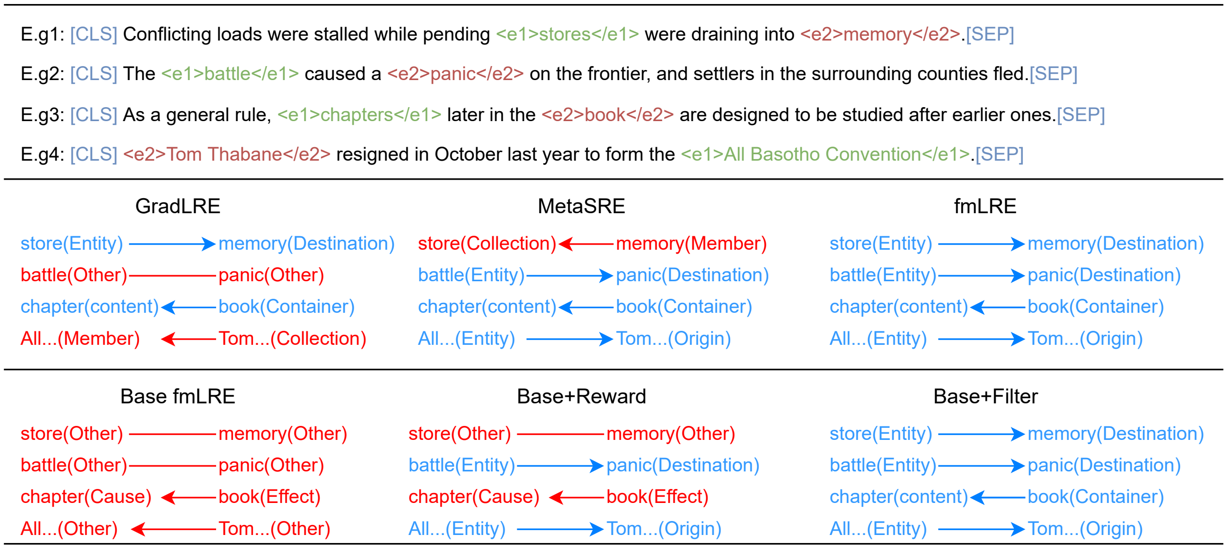
Ablation Experiments

Iterative Feedback As shown in Table 4, after adding the feature mapping space to calculate the similarity, the performance of fmLRE has greatly improvement compared with the basic model. One reason is the positive impact of iterative feedback. Another reason is that fmLRE filters high-precision pseudo-labels. Namely, it uses the mapping space to classify and calculate the similarity of each pseudo-label, not only has the category information of each relation, but also can measure the similarity more accurately, adjusting according to the characteristics of different datasets.

Filter Mechanism The improvement brought by the filter mechanism is not very stable, and it plays a role of reducing the negative impact of the noise data. Therefore, it can be seen that it has a greater effect on the dataset with many unknown relations like TACRED. In the training process, special data is also one of the important factors that have inconsistent effects on the model. The noise filter mechanism

Dataset	SemEval(%)			TACRED(%)		
Ratio	5%	10%	30%	3%	10%	15%
Only Self-train	71.68	74.25	75.52	44.31	53.76	55.73
+fm Similarity	80.94	82.04	83.08	47.93	57.54	58.29
+Filter	**82.11**	**83.83**	**85.97**	**50.83**	**58.94**	**60.62**
Model+BERT	82.11	83.83	**85.97**	50.83	58.94	60.62
Model+MTB	**83.40**	**84.12**	85.94	**54.66**	**64.23**	**61.91**

Table 4: Performance comparison of different pre-trained models or different modules. The upper part is the performance after the model gradually adds important modules, and the lower part is the performance comparison of the model using different pre-trained models


E.g1: [CLS] Conflicting loads were stalled while pending <e1>stores</e1> were draining into <e2>memory</e2>.[SEP]

E.g2: [CLS] The <e1>battle</e1> caused a <e2>panic</e2> on the frontier, and settlers in the surrounding counties fled.[SEP]

E.g3: [CLS] As a general rule, <e1>chapters</e1> later in the <e2>book</e2> are designed to be studied after earlier ones.[SEP]

E.g4: [CLS] <e2>Tom Thabane</e2> resigned in October last year to form the <e1>All Basotho Convention</e1>.[SEP]

GradLRE
store(Entity) → memory(Destination)
battle(Other) — panic(Other)
chapter(content) ← book(Container)
All...(Member) ← Tom...(Collection)
MetaSRE
store(Collection) ← memory(Member)
battle(Entity) → panic(Destination)
chapter(content) ← book(Container)
All...(Entity) → Tom...(Origin)
fmLRE
store(Entity) → memory(Destination)
battle(Entity) → panic(Destination)
chapter(content) ← book(Container)
All...(Entity) → Tom...(Origin)
Base fmLRE
store(Other) — memory(Other)
battle(Other) — panic(Other)
chapter(Cause) ← book(Effect)
All...(Other) ← Tom...(Other)
Base+Reward
store(Other) — memory(Other)
battle(Entity) → panic(Destination)
chapter(Cause) ← book(Effect)
All...(Entity) → Tom...(Origin)
Base+Filter
store(Entity) → memory(Destination)
battle(Entity) → panic(Destination)
chapter(content) ← book(Container)
All...(Entity) → Tom...(Origin)

Figure 4: Experimental case analysis. In the examples we use different colors to label entities and special tokens, and in the prediction results, we use blue to indicate that the prediction is correct, and red indicates wrong.

is based on the number of pseudo-label eliminations and still has limitations: (1) Data may be mistakenly filtered with the increase of elimination times, reducing the diversity of the data; (2) The hyperparameter setting needs to be manually tuned or selected by validation set. Therefore, self-adaptive learning may be a further improving way.

Embedding Layer As mentioned before, BERT could be replaced with the pre-trained model MTB (Soares et al. 2019) to further improve the performance upper limit of fmLRE. According to the experimental results, MTB is beneficial to the LRE task. In addition, it helps to reach convergence faster and has higher efficiency in the training process.

Case Study

We further compare fmLRE with other baselines through case studies. Due to the space limitation, we only compare fmLRE with GradlRE and MetaSRE. The instances of the module is consistent with the setting in the ablation experiments. In addition, it tends to be predicted as `unknown` when the model makes a wrong prediction, which has similar phenomenon in module comparison. Three examples are presented in the two comparisons and illustrated in Figure 4. First, in the model comparison, since the performance of each model is similar, there is no obvious difference in these cases. However, the performance of fmLRE is more stable encountering random samples compared to its counterparts. Second, in the module comparison, it can be clearly seen that the performance of the model is improved after using the two modules. The reason why the model predicts incorrectly and always predicts as `unknown` is that it has not learned the useful features of this relation category. Such phenomenon is more frequent on TACRED with many unknown relations.

Conclusions

This paper proposes a positive iterative feedback mechanism based on the self-training, using a combination of reinforcement learning and metric learning, fmLRE, to calculate the similarity between the support and query sets and to alleviate the selection bias. The limitations of fmLRE model are two-fold: First, it has no advantage in generalization performance. Second, fmLRE cannot solve the zero-shot problem. The generalization and more few-shot learning methods will be studied in the future improvement work.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. The work was supported in part by the 13th Five-Year All-Army Common Information System Equipment Pre-Research Project (Grant Nos. 31514020501, 31514020503). All opinions are of the authors and do not reflect the view of sponsors.

References

Bach, N.; and Badaskar, S. 2007. A Review of Relation Extraction. In *International Conference on Mechanical, Control and Computer Engineering*.

Brin, S. 1998. Extracting Patterns and Relations from the World Wide Web. In *Proceedings of the International Workshop on the World Wide Web and Databases*.

Cetoli, A. 2020. Exploring the zero-shot limit of FewRel. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Chen, J.; Ji, D.; Chew, L. T.; and et al. 2006. Relation Extraction Using Label Propagation Based Semi-Supervised Learning. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Deng, S.; Zhang, N.; and Chen, H. 2022. Knowledge Extraction in Low-Resource Scenarios: Survey and Perspective. In *arXiv:2202.08063*.

Devlin, J.; Chang, M.; Kenton, L.; and et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Dong, M.; Pan, C.; and Luo, Z. 2021. MapRE: An Effective Semantic Mapping Approach for Low-resource Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Gupta, P.; Roth, B.; and Schütze, H. 2018. Joint Bootstrapping Machines for High Confidence Relation Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Han, J.; Cheng, B.; and Lu, W. 2021. Exploring Task Difficulty for Few-Shot Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Han, X.; Zhu, H.; Yu, P.; and et al. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hendrickx, I.; Su, N. K.; Kozareva, Z.; and et al. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hu, L.; Zhang, L.; Shi, C.; and et al. 2019. Improving Distantly-Supervised Relation Extraction with Joint Label Embedding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Hu, X.; Zhang, C.; Ma, F.; and et al. 2021a. Semi-supervised Relation Extraction via Incremental Meta Self-Training. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Hu, X.; Zhang, C.; Yang, Y.; and et al. 2021b. Gradient Imitation Reinforcement Learning for Low Resource Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Levy, O.; Seo, M.; Choi, E.; and et al. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*.

Li, G.; Chen, X.; Wang, P.; Xie, J.; and Luo, Q. 2022. FastRE: Towards Fast Relation Extraction with Convolutional Encoder and Improved Cascade Binary Tagging Framework. In *Proceedings of the 31st International Joint Conference On Artificial Intelligence*.

Li, W.; and Qian, T. 2020. Exploit Multiple Reference Graphs for Semi-supervised Relation Extraction. In *ArXiv: abs/2010.11383*.

Liang, X.; Wu, S.; Li, M.; and et al. 2022. Modeling Multi-Granularity Hierarchical Features for Relation Extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lin, Y.; Shen, S.; Liu, Z.; and et al. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Mintz, M.; Bills, S.; Snow, R.; and et al. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

Pawar, S.; Palshikar, K. G.; and Bhattacharyya, P. 2017. Relation Extraction : A Survey. In *ArXiv: abs/1712.05191*.

Peng, H.; Gao, T.; Han, X.; and et al. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Qu, M.; Gao, T.; Xhonneux, L.-P. A. C.; and et al. 2020. Few-Shot Relation Extraction via Bayesian Meta-Learning on Relation Graphs. In *Proceedings of the 37th International Conference on Machine Learning*.

Ren, H.; Cai, Y.; Chen, X.; and et al. 2020. A Two-phase Prototypical Network Model for Incremental Few-shot Relation Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Rosenberg, C.; Hebert, M.; and Schneiderman, H. 2005. Semi-Supervised Self-Training of Object Detection Models. In *Seventh IEEE Workshops on Applications of Computer Vision*.

Soares, L. B.; FitzGerald, N.; Ling, J.; and et al. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sung, F.; Yang, Y.; Zhang, L.; and et al. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Vashishth, S.; Joshi, R.; Prayaga, S. S.; and et al. 2018. RESIDE: Improving Distantly-Supervised Neural Relation Extraction using Side Information. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Wu, S.; and He, Y. 2019. Enriching Pre-Trained Language Model with Entity Information for Relation Classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Xu, Y.; Mou, L.; and Li, G. 2015. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Yao, L.; Haghighi, A.; Riedel, S.; and et al. 2011. Structured Relation Discovery using Generative Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yuan, C.; and Eldardiry, H. 2021. Unsupervised Relation Extraction: A Variational Autoencoder Approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115.

Zhang, D.; and Wang, D. 2015. Relation Classification via Recurrent Neural Network. In *ArXiv: abs/1508.01006*.

Zhang, N.; Xu, X.; Tao, L.; and et al. 2022. DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population. In *arXiv preprint arXiv:2201.03335*.

Zhang, Y.; Zhong, V.; Chen, D.; and et al. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.